Social Sparsity! Neighborhood Systems Enrich Structured Shrinkage Operators

Matthieu Kowalski, Kai Siedenburg, Monika Dörfler

Social Sparsity! Neighborhood Systems Enrich Structured Shrinkage Operators

Matthieu Kowalski, Kai Siedenburg, Monika Dörfler

***Abstract*—Sparse and structured signal expansions on dictionaries can be obtained through explicit modeling in the coefficient domain. The originality of the present article lies in the construction and the study of generalized shrinkage operators, whose goal is to identify structured significance maps and give rise to structured thresholding. These generalize Group Lasso and the previously introduced Elitist Lasso by introducing more flexibility in the coefficient domain modeling, and lead to the notion of *social sparsity*. The proposed operators are studied theoretically and embedded in iterative thresholding algorithms. Moreover, a link between these operators and a convex functional is established. Numerical studies on both simulated and real signals confirm the benefits of such an approach.**

***Index Terms*—Structured Sparsity, Iterative Thresholding, Convex Optimization**

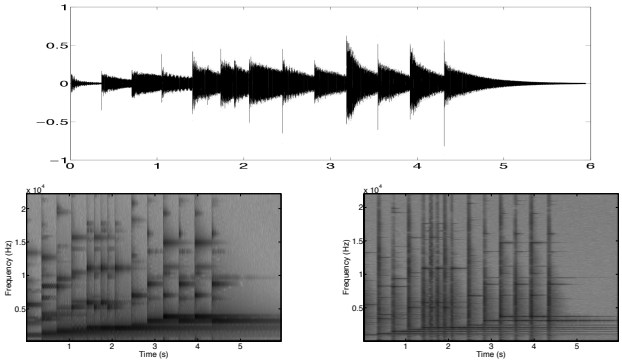

Fig. 1. Time-frequency images. Top: signal samples, bottom-left: representation adapted to transients. Bottom-right, representation adapted to tonals.

I. Introduction

A wide range of inverse problems arising in signal processing have benefited from *sparsity*. Introduced in the mid 90's by Chen, Donoho and Saunders [1], the idea is that a signal can be efficiently represented as a linear combination of elementary atoms chosen from an appropriate *dictionary*. Here, *efficiently* may be understood in the sense that only few atoms are needed to reconstruct the signal. The same idea appeared in the machine learning community [2], where often only few variables are relevant in inference tasks based on observations living in very high dimensional spaces.

The natural measure of the cardinality of a support set, and hence its sparsity, is the ℓ_0 "norm" which counts the number of non-zero coefficients. Minimizing such a penalty leads to a combinatorial problem which is usually relaxed into a ℓ_1 norm which is convex.

Solving an inverse problem by using the sparse principle can be done by the following steps:

- Choose a dictionary where the signal of interest is supposed to be sparse. Such a choice is driven by the nature of the signal: Gabor dictionaries (for audio signals for example), wavelet dictionaries (for images) are commonly used, among others. The dictionary can even be learned directly on a class of signals [3]. In order to be able to use the sparse principle, this step of choosing an appropriate dictionary is obviously crucial.
- Choose a loss in order to link the observations, or measured signals, to the sought signals. While other loss functions, such as the logistic loss, may be used, in the current contribution, we focus on the classical ℓ_2 norm used with success in various problems.
- Apply an ℓ_1 penalty on the coefficients of the signal expanded in the dictionary.

The resulting convex optimisation problem is known as the *Basis Pursuit (Denoising)* [1] or the *Lasso* [2]. This approach can be viewed as a *synthesis* model of the signal: one directly estimates its coefficients inside a dictionary in order to synthesize the signal from these coefficients.

One of the main limitations of this approach to sparse modeling is that all the coefficients are treated independently. Most natural signals are highly structured, however, and the structures which become visible in an *analysis* of a signal correspond to the physical prior which could be used for its processing. We instance such an observation on an audio signal.

A. From sparsity to the need of structures

Fig. 1 displays the time samples of a glockenspiel signal, and two time-frequency representations using a modified discrete cosine transform (MDCT), one with a narrow band analysis window adapted for the tonal part (well localized in frequency) and one with a large band analysis window

M. Kowalski is with Laboratoire des Signaux et Systèmes, UMR 8506 CNRS - SUPELEC - Univ Paris-Sud, 91192 Gif-sur-Yvette Cedex, France (e-mail: matthieu.kowalski@lss.supelec.fr).

K. Siedenburg was with Austrian Institute for Artificial Intelligence (OFAI), Freyung 6/6, A-1010 Vienna, Austria. He is now at the Schulich School of Music, McGill University, Montreal, Canada (e-mail: kai.siedenburg@mail.mcgill.ca)

M. Dörfler is with Numerical Harmonic Analysis Group, Faculty of Mathematics, University of Vienna, Alserbachstrasse 23, 1090 Wien, Austria (e-mail: monika.doerfler@univie.ac.at)

This work was supported by the Austrian Science Fund (FWF) : T384-N13 and the WWTF project Audio-Miner (MA09-024).

(well localized in time). Clearly, the time samples are not a sparse representation. The two time-frequency representations display only few "big" coefficients (in dark gray), while the organisation of these coefficients (their structure) depends on the choice of the basis. Hence an idea is to construct a dictionary as a union of two others, each adapted to the "morphological layer". Such an approach has been proposed in [4] as *hybrid model* for audio signals, and in [5] as *morphological* model for images. A more theoretical study has been performed in [6], where sufficient conditions that guarantee the uniqueness of a sparse representation in union of orthogonal bases were obtained.

In addition to these observations, we notice a grouping effect of the coefficients in both time- and frequency-direction, for each of the dictionaries used. The main motivation for the work presented in this article is to better understand how this grouping effect can be taken into account in this and similar situations, in order to obtain a more reliable sparse representation in a corresponding dictionary.

Our main contribution is to propose the concept of *social sparsity*: a certain, possibly weighted, neighborhood of a given coefficient is considered for deciding whether to keep or discard the coefficient under consideration. This idea was first introduced in [7]; it was equipped with weights and evaluated for various audio applications in [8]. For the realization of the intuitive idea that a coefficient's neighborhood should be relevant for its impact, we construct *structured shrinkage* operators which are directly derived from classical proximity/shrinkage operators such as the Group-Lasso. However, while the classical proximity operators are directly linked to convex regression problems with mixed norm priors on the coefficients, the new, structured, shrinkage operators can not be directly linked to a convex minimization problem. While the convergence of related iterative algorithms for the classical shrinkage operators and their generalizations was studied in [9] in a rather general setting, the theoretical properties of the new operators have not been considered so far. In the current contribution, we establish a formal relation between the structured shrinkage operators and the minimization of a convex functional by introducing an *expansion operator*, which maps the coefficient space into a higher-dimensional space. Exploiting this extension, the shrinkage operators are linked to a related convex problem, whose convergence properties are known. While proving convergence of the initial algorithm associated with the new shrinkage operators remains an open problem, numerical experiments show that its behavior is sufficiently similar to the behavior of the algorithm derived from the more formal convex formulation. By replacing an oblique by an orthogonal projection, we also propose another alternative operator, for which the convergence to a fixed point is warranted. Our framework also allows the inclusion of the recently introduced Latent-Group-Lasso [10], [11], to whose performance the new algorithms will also be compared.

B. Outline

Section II introduces the mathematical framework used for this article and Section III presents the state of the art related to this framework. The structured shrinkage operators are introduced in Section IV where their theoretical study is derived. We show in Section V some practical implementation of our approach and present numerical results of its performance in denoising tasks on audio and image-signals.

II. Mathematical framework

This section introduces notation used throughout the paper as well as some useful results from convex analysis.

A. Notation

We will denote the observed signal as $\mathbf{y} \in \mathbb{R}^L$, obtained from the signal of interest $\mathbf{s} \in \mathbb{R}^L$ corrupted by an additive noise $\mathbf{b} \in \mathbb{R}^L$, i.e.

$$\mathbf{y} = \mathbf{s} + \mathbf{b} \ .$$

The matrix of the dictionary is denoted by $\mathbf{\Phi} \in \mathbb{C}^{L,N}$ and the synthesis coefficients of $\mathbf{s}$ in $\mathbf{\Phi}$ are denoted by $\boldsymbol{\alpha} \in \mathbb{C}^N$, such that

$$\mathbf{y} = \mathbf{s} + \mathbf{b} \ = \mathbf{\Phi}\boldsymbol{\alpha} + \mathbf{b} \ .$$

A sparse estimation of $\mathbf{s}$ is given by the Lasso [2] or Basis Pursuit Denoising [1]:

$$\hat{s} = \mathbf{\Phi} \operatorname*{argmin}_{\alpha \in \mathbb{C}^N} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \ , \quad \lambda > 0 \ . \quad (1)$$

In this article we choose to use the general convex formulation

$$\hat{\mathbf{s}} = \mathbf{\Phi} \operatorname*{argmin}_{\alpha \in \mathbb{C}^N} \frac{1}{2}\|\mathbf{y} - \mathbf{\Phi}\boldsymbol{\alpha}\|_2^2 + \lambda\Omega(\boldsymbol{\alpha}) \quad (2)$$

where Ω is a convex penalty. Depending upon the choice made for Ω, different kinds of sparsity or structure can be enforced.

Remark 1. *We choose to limit our purpose to the case of* sparse regression*, where the synthesis coefficients $\boldsymbol{\alpha}$ of the signal of interest $\mathbf{s}$ are estimated from a single measurement $\mathbf{y}$ only corrupted by an additive noise. However, this approach can be extended to more general inverse problems where several signals have to be estimated from several measurements such as in source separation [12].*

Remark 2. *The functionals appearing in* (1) *and* (2) *are convex but not necessarily strictly convex. Then, the set of minimizers is not necessarily a singleton. However, with a slight abuse of notation, we choose the notation* argmin *to represent any minimizer, as the choice of a particular minimizer has no consequences for the rest of the paper. One can refer to [13] and [14] for discussions of the uniqueness of the ℓ_1 problem.*

B. Short reminder of Convex optimization

The algorithms proposed in this paper are issued from convex optimization methods and rely on the notion of the *proximity operator*, introduced by Moreau [15], which allows to deal with non-smooth functionals.

Definition 1 (Proximity operator)**.** *Let $\varphi : \mathbb{C}^N \to \mathbb{C}^N$ be a lower semicontinuous, convex function. The* proximity operator *of φ denoted by $\mathrm{prox}_\varphi : \mathbb{C}^N \to \mathbb{C}^N$ is given by*

$$\mathrm{prox}_\varphi(\mathbf{z}) = \underset{\mathbf{u}\in\mathbb{C}^N}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{z}-\mathbf{u}\|_2^2 + \varphi(\mathbf{u}). \tag{3}$$

The most well-known example of such an operator is the shrinkage given by the ℓ_2 Tikhonov regularization and the soft-thresholding given by the ℓ_1 norm.

If one is able to compute the proximity operator of a convex regularizer Ω, then the minimizer of the convex functional (2) can be obtained by using proximal algorithms. The simplest proximal algorithm was found in the ℓ_1 case by several researchers using very different approaches. In [16] Daubechies and coauthors derived the thresholded Landweber iterations using a surrogate and proved the convergence using Opial's fixed point Theorem. In [17], Figueiredo *et al.* found the same algorithm thanks to an expectation/maximization formulation. A more general version using the proximity operators was given by the forward-backward algorithm studied by Combettes *et al.* [18]. We will refer to this algorithm as the Iterative Shrinkage/Thresholding Algorithm (ISTA) as in [19] and we restate it in Algorithm 1 for the problem studied here.

Algorithm 1: ISTA

Initialization: $\boldsymbol{\alpha}^{(0)} \in \mathbb{C}^N$, $k=1$, $\gamma = \|\boldsymbol{\Phi}\boldsymbol{\Phi}^*\|$
repeat
  $\boldsymbol{\alpha}^{(k)} = \mathrm{prox}_{\frac{\lambda}{\gamma}\Omega}\left(\boldsymbol{\alpha}^{(k-1)} + \frac{1}{\gamma}\boldsymbol{\Phi}^*(\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha}^{(k-1)})\right)$;
  $k = k+1$;
until *convergence*;

ISTA can be viewed as a generalization of the gradient descent for the non smooth functional (2). The algorithm is very simple, but converges slowly in practice. Recent advances in convex optimization lead to more efficient algorithms, we refer to [20] for a thorough discussion of proximal algorithms and their accelerations. Algorithm 2 describes the Fast Iterative Shrinkage/Thresholding Algorithm as proposed in [19]. The

Algorithm 2: FISTA

Initialization: $\boldsymbol{\alpha}^{(0)} \in \mathbb{C}^N$, $k=1$, $\gamma = \|\boldsymbol{\Phi}\boldsymbol{\Phi}^*\|$,
$\mathbf{z}^{(0)} = \boldsymbol{\alpha}^{(0)}$, $\tau^{(0)} = 1$.
repeat
  $\boldsymbol{\alpha}^{(k)} = \mathrm{prox}_{\frac{\lambda}{\gamma}\Omega}\left(\mathbf{z}^{(k-1)} + \frac{1}{\gamma}\boldsymbol{\Phi}^*(\mathbf{y} - \boldsymbol{\Phi}\mathbf{z}^{(k-1)})\right)$;
  $\tau^{(k)} = \frac{1}{2}\left(1+\sqrt{1+4{\tau^{(k-1)}}^2}\right)$;
  $\mathbf{z}^{(k)} = \boldsymbol{\alpha}^{(k)} + \frac{\tau^{(k-1)}-1}{\tau^{(k)}}(\boldsymbol{\alpha}^{(k)} - \boldsymbol{\alpha}^{(k-1)})$;
  $k = k+1$
until *convergence*;

choice $\gamma = \|\boldsymbol{\Phi}\boldsymbol{\Phi}^*\|$ is a sufficient condition in order to ensure the convergence of ISTA and FISTA. In some cases, it can be useful to perform a line search for γ at each iteration (see [19]). However, we observed that when the matrix $\boldsymbol{\Phi}\boldsymbol{\Phi}^*$ is not "too badly conditioned", such a line search does not bring any computational advantage. In particular in the experiments performed in Section V, the default constant choice for γ works well.

Having introduced practical algorithms to deal with convex functionals as in (2), in the next section, we turn to reviewing some state-of-the-art approaches that go beyond the simple sparsity paradigm.

III. State of the art

Considering grouping structures of coefficients appears as a natural idea in the sparse regression context. A simple way to obtain such groupings is the use of mixed norms, which allows to regroup coefficients. We first give the definition of mixed norms and their proximity operators which will be used later. Other kinds of grouping structures which appear in the literature are presented afterwards.

A. Mixed norms

Mixed norms were introduced by Benedek and Panzone [21] in the early 1960's in mathematics.

1) Definition on two levels: We give here the general definition as in [7], [9].

Definition 2 (Two-level mixed norms)**.** *Let $\mathbf{x} \in \mathbb{R}^N = \mathbb{R}^{G\times M}$ be indexed by a double index $(g,m) \in \mathbb{N}^2$ such that $\mathbf{x} = (x_{g,m})$.*

Let $p, q \geq 1$, and $\mathbf{w} \in \mathbb{R}^N_{+,*}$ be a sequence of strictly positive weights labeled by double index (g,m). We call $\ell_{\mathbf{w};p,q}$ the mixed norm *of $\mathbf{x} \in \mathbb{R}^N$ defined by*

$$\|\mathbf{x}\|_{\mathbf{w};p,q} = \left(\sum_{g=1}^{G}\left(\sum_{m=1}^{M} w_{g,m}|x_{g,m}|^p\right)^{q/p}\right)^{1/q}.$$

The cases $p=+\infty$ and $q=+\infty$ are obtained by replacing the corresponding sum by the supremum.

Two mixed norms appear quite naturally by playing with the different values of p and q: the ℓ_{21} and ℓ_{12} norms. The ℓ_{21} norm was used with the name Group-Lasso [22] (G-Lasso) in machine learning, but also Multiple Measurement Vectors [23] or joint sparsity [24] in signal processing. In the context of regression, the main aim of such a norm is to keep or discard entire groups of coefficients. Indeed, if we consider the special case of an orthogonal basis, only the most energetic groups remain.

The ℓ_{12} norm was introduced under the name of Elitist-Lasso [7], [9] (E-Lasso), and latter called Exclusive Lasso in [25]. With such a penalty, and if $\boldsymbol{\Phi}$ is an orthogonal basis, we keep the biggest coefficients relative to the others . Such behavior can be expected in applications such as source separation [12].

2) Extension to 3 levels: This notion of mixed norms can be extended to more than two levels. On three levels, the definition becomes [26]

Definition 3 (Three-level mixed norms)**.** *Let $\mathbf{x} \in \mathbb{R}^N = \mathbb{R}^{K\times G\times M}$ be indexed by a triple index $(k,g,m) \in \mathbb{N}^3$ such that $\mathbf{x} = (x_{k,g,m})$. Let $p,q,r \geq 1$ and $\mathbf{w} \in \mathbb{R}^N_{+,*}$ a sequence*

of strictly positive weights. We call $\ell_{\mathbf{w};p,q,r}$ *the* mixed norm *of* $\mathbf{x}$ *defined by*

$$\|\mathbf{x}\|_{\mathbf{w};pqr} = \left(\sum_{k=1}^{K}\left(\sum_{g=1}^{G}\left(\sum_{m=1}^{M} w_{k,g,m}|x_{k,g,M}|^p\right)^{q/p}\right)^{r/q}\right)^{1/r}.$$

The cases $p=+\infty$, $q=+\infty$ *and* $r=+\infty$ *are obtained by replacing the corresponding sum by the supremum.*

The ℓ_{212} mixed norm was used with success in Magnetoencephalography inverse problems [26] and could be called Elitist-Group-Lasso (EG-Lasso).

3) Proximity Operators: In order to optimize certain convex problems using mixed norms, their proximity operator needs to be computed. The following proposition summarizes the different operators for various norms, cf. [9].

Proposition 1 (Proximity operators for mixed norms)**.** *Let* $\mathbf{x}\in\mathbb{R}^N$ *and* $\mathbf{z}\in\mathbb{R}^N$. *Let* $\mathbf{w}\in\mathbb{R}^N_{+,*}$ *be a vector of weights. We suppose that* $\mathbf{x},\mathbf{z}$ *and* $\mathbf{w}$ *are indexed by* (g,m).

G-Lasso $\ell_{\mathbf{w};21}$ ***norm****: In this case, the vector of weights* $\mathbf{w}$ *is used to weight each group, i.e.* $\forall m, w_{g,m}=w_g$.

The proximity operator associated to the $\ell_{\mathbf{w};21}$ *norm is given by* $\mathbf{x}=\mathrm{prox}_{\lambda\|\cdot\|_{\mathbf{w},21}}(\mathbf{z})$ *where* $\mathbf{x}$ *reads for each coordinate:*

$$x_{g,m} = z_{g,m}\left(1-\frac{\lambda\sqrt{w_g}}{\|\mathbf{z}_g\|_2}\right)^+,$$

and $\mathbf{z}_g$ *is the vector formed by the coefficients indexed by* m.

E-Lasso $\ell_{\mathbf{w};12}$ ***norm****: Let* $r_{g,m}\overset{\text{def}}{=}|z_{g,m}|/w_{g,m}$ *and for each* g, *let the indexing denoted by* m'_g *be defined such that* $\forall m'_g, r_{g,m'_g+1}\leq r_{g,m'_g}$ *and re-order the* $z_{g,m}$ *according to this index. Let the index* M_g *be such that:*

$$\lambda\sum_{m'_g=1}^{M_g} w^2_{g,m'_g}\left(r_{g,m'_g}-r_{g,M_g}\right) < r_{g,M_g} \leq \lambda\sum_{m'_g=1}^{M_g+1} w^2_{g,m'_g}\left(r_{g,m'_g}-r_{g,M_g}\right)$$

The proximity operator $\mathbf{x}=\mathrm{prox}_{\lambda\|\cdot\|_{\mathbf{w},12}}(\mathbf{z})$ *is given coordinate-wise:*

$$x_{g,m}=\frac{z_{g,m}}{|z_{g,m}|}\left(|z_{g,m}|-\frac{\lambda}{1+\lambda K_{\mathbf{w}_g}}\sum_{m'_g=1}^{M_g}|z_{g,m'_g}|\right)^+,$$

where $K_{\mathbf{w}_g}=\sum_{m'_g=1}^{M_g} w^2_{g,m'_g}$.

EG-Lasso $\ell_{\mathbf{w};212}$ ***norm****: Let* $\mathbf{x}$ *be indexed by* (h,g,m). *Let* $\mathbf{w}\in\mathbb{R}^N$ *be a vector of positive weights such that* $\forall m,\ w_{h,g,m}=w_{h,g}$. *Let us define* $r_{h,g}\overset{\text{def}}{=}\|\mathbf{z}_{h,g}\|_2/\sqrt{w_{h,g}}$. *For each* h, *let the indexing denoted by* g'_h *be defined such that* $\forall g'_h, r_{h,g'_h+1}\leq r_{h,g'_h}$. *Let the index* G_h *be such that:*

$$\lambda\sum_{g'_h=1}^{G_h} w_{h,g'_h}\left(r_{h,g'_h}-r_{h,G_h}\right) < r_{h,G_h} \leq \lambda\sum_{g'_h=1}^{G_h+1} w_{h,g'_h}\left(r_{h,g'_h}-r_{h,G_h}\right).$$

Denoting by $[\mathbf{z}_{h,g'_h}]=\sqrt{w_{h,g'_h}}\|\mathbf{z}_{h,g'_h}\|_2$ *and supposing that they are ordered by* g'_h, *then* $\mathbf{x}=\mathrm{prox}_{\lambda\|\cdot\|^2_{\mathbf{w};212}}(\mathbf{z})$ *is given, for each coordinate* (h,g,m), *by:*

$$x_{h,g,m}=z_{h,g,m}\left(1-\frac{\lambda\sqrt{w_{h,g}}}{1+\lambda K_{\mathbf{w}_h}}\frac{\sum_{g'_h=1}^{G_h}[\mathbf{z}_{h,g'_h}]}{\|\mathbf{z}_{h,g}\|_2}\right)^+,$$

where $K_{\mathbf{w}_h}=\sum_{g_h=1}^{G_h} w_{h,g_h}$.

Remark 3. *The proximity operators of the* ℓ_{121} *mixed norms and of general mixed norms defined on more than three levels are not computable in a closed form. In fact, one needs to compute a "Group-Lasso" proximity operator with weights varying in groups, which does not admit a closed form.*

It is interesting to note that in [27] a hierarchical formulation of the dependencies leads to a $\ell_{1,\frac{4}{3}}$ mixed norm.

Notice that the mixed norms as defined here do not consider any overlap between the groups. The need for overlapping groups was recognized by many authors, see [10], [28]–[30], and different strategies have been proposed.

B. A step beyond mixed norms

In [10], [11], starting from the observation that the Group-Lasso *discards* all coefficients in a given group, the authors define a new norm in order to deal with overlapping groups: the Latent-Group-Lasso. This definition of a new convex penalty leads to the desired results: all coefficients belonging to the same group are kept, even if they also belong to another group which is discarded. The remaining support is thus a union of groups instead of the complement of a union as in the Group-Lasso. However, in general, there is no closed form for the proximity operators corresponding to the Latent-Group-Lasso. The authors propose a reformulation and solution of the convex problem by introducing a latent variable in a high dimensional space through the duplication of the variables belonging to overlapping groups.

In the particular case where the groups are all the subsets of a given cardinality, the proximity operator can be computed exactly. This particular case corresponds in fact to the so called k–support norm [31], which is closely related to the elastic net [32]. Furthermore, iterative algorithms exist, cf. [33] if one needs to compute the proximity operator in the general case.

Despite the "discarding" behavior of the Group-Lasso, mixed norms with overlaps have been studied in [29]. Again, the proximity operator has no closed form, but an iterative scheme is proposed. The mixed norm with overlaps corresponds actually to a particular case of the regularizer proposed in [30], where a partition function is introduced to construct a convex penalty.

As we will see in Section IV-C2, all these methods are closely related. The study of the proposed structured shrinkage operators naturally leads to convex functionals which correspond to the problems proposed in the previously mentioned contributions.

Various other kinds of structures have been proposed for refining the model of group based sparsity. For example, in [34] a hierarchy on groups was introduced. Such a behavior allows for sparsity inside the group, in addition to sparsity between the groups. In particular, their hierarchical sparse coding included the sums of convex penalties such as a $\ell_{21} + \ell_1$ composite norm, also known as the HiLasso [35]. Such a composite norm was used with success for Magnetoencephalography inverse problems with respect to time-frequency dictionaries [36]. In [37], the authors studied a very general mixed norm, allowing to generalize the Group-Lasso and the hierarchical sparse coding.

In [38], the authors propose a family of *"convex penalty functions, which encode this prior knowledge by means of a set of constraints on the absolute values of the regression coefficients"*. In practice, the structure is encoded by means of an auxiliary variable. This formulation is general enough to obtain the Lasso and the Group-Lasso as special cases. The drawback of this flexibility may lie in the difficulty to define the desired structure, and the computational complexity of optimization when the problem lives in a high dimensional space.

In addition to the convex approaches, several other solutions were proposed for the *structured sparsity* problem. Among others, we can cite the model-based compressive sensing [39], and approaches based on coding theory [40] or Bayesian methods (see e.g. [41] and references therein).

IV. Structured shrinkage

One of the main shortcomings of the various "structured block sparsity" approaches exposed above, is that the definition of the groups must be done *a priori*. However, in many situations, we just have a general idea of the grouping structure, and fixing the groups can be too rigid.

Instead of defining groups, and therewith keeping or discarding entire blocks of coefficients, a notion of neighborhood-based selection was proposed in [42]. The introduction of this neighborhood gives rise to *"social sparsity"*: a decision can be made coefficient by coefficient by taking into account the "weight" of a coefficient's neighborhood. The latter still has to be defined a priori, but the possibility of overlap between neighborhoods instead of groups relaxes the rigor of the (Group-)Lasso approaches. For application, neighborhoods then should be chosen according to the grouping structures observed in the specific signal class under observation, as e.g. the persistence of tonal and transient parts in audio signals noted above or the father-son persistence in wavelet expansions of images to be addressed below.

In order to present this approach, we first give the definition of the neighborhood and re-state the shrinkage operators empirically introduced in [42] and equipped with weights in [8].

A. Structured shrinkage operators

To exploit structures in the synthesis coefficients, (like persistence in time or frequency in audio signals as in Fig. 1 in the introductory Section I), we will refine some classical shrinkage operators by taking into account the neighborhood of a coefficient. To an index k in a set $\mathcal{I}$, we associate a weighted neighborhood $\mathcal{N}(k) = \{k' \in \mathcal{I} \ : \ w_{k'}^{(k)} \neq 0\}$ with weights $w_{k'}^{(k)}$ such that $w_{k'}^{(k)} \geq 0$ for all $k' \in \mathcal{I}$, $w_k^{(k)} > 0$ and $\sum_{k' \in N(k)} {w_{k'}^{(k)}}^2 = 1$. This notion of neighborhood is illustrated on Fig. 2.

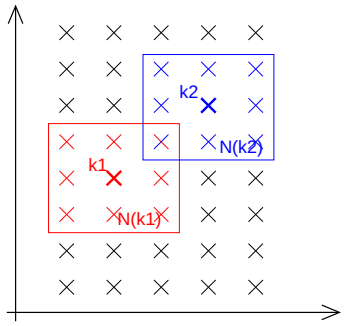

Fig. 2. The neighborhood of the coefficient k_1 is given by the red window, and the neighborhood of the coefficient k_2 by the blue one. These two neighborhoods share one coefficient. When considering the red group, coefficients are weighted by some weights $w_{k'}^{k_1} > 0$, $k' \in \mathcal{N}(k_1)$. Outside the red group, the weights are equal to zero. When considering the blue group, coefficients are weighted by some weights $w_{k'}^{k_2} > 0$, $k' \in \mathcal{N}(k_2)$.

Once the neighborhood is defined, we can define shrinkage operators on it. These operators are constructed with the shrinkage operators given by the proximity operator corresponding to the Group/ Elitist/ Elitist-Group-Lasso defined in Section III, Proposition 1, by considering the "groups" formed by the neighborhood.

1) WG-Lasso: We first introduce the Windowed-Group-Lasso [42] shrinkage operator, defined as

$$\begin{aligned} \mathbb{S}_\lambda^{wgl}(\boldsymbol{\alpha}) : \mathbb{C}^N &\to \mathbb{C}^N \\ \boldsymbol{\alpha} &\mapsto \underline{\boldsymbol{\alpha}} \end{aligned}$$

such that for all k ,

$$\underline{\alpha}_k = \alpha_k \left(1 - \frac{\lambda}{\sqrt{\sum_{k' \in \mathcal{N}(k)} w_{k'}^{(k)} |\alpha_{k'}|^2}} \right)^+ . \qquad (4)$$

The idea of this shrinkage operator is to select a coefficient if the energy of its neighborhood is sufficiently large. Consequently, an isolated "big" coefficient can be discarded, but a "small" coefficient in the middle of big ones can be kept. Such a notion of neighborhood can also be found earlier in [43], where a similar thresholding rule was studied in the context of SURE wavelet estimation.

2) WE-Lasso: Instead of considering a positive correlation between the coefficient in the neighborhood, one can consider a negative correlation as in the Elitist-Lasso. This leads to the following shrinkage operator, which we will call the *Windowed-Elitist-Lasso*. For each neighborhood $\mathcal{N}(k)$, let the indexing denoted by k' be defined such that $\forall k' \in \mathcal{N}(k), w_{k'+1}^{(k)} |\alpha_{k'+1}| \leq w_{k'}^{(k)} |\alpha_{k'}|$. Let the index K_k be such

that:

$$\lambda \sum_{k'=1}^{K_k} \left(w_{k'}^{(k)} \alpha_{k'} - w_{K_k}^{(k)} \alpha_{K_k} \right) < \alpha_{K_k} \leq \lambda \sum_{k'=1}^{K_k+1} \left(w_{k'}^{(k)} \alpha_{k'} - w_{K_k}^{(k)} \alpha_{K_k} \right)$$

Then the Windowed-Elitist-Lasso shrinkage operator is given by

$$\begin{aligned} \mathbb{S}_{\lambda}^{wel}(\boldsymbol{\alpha}) : \mathbb{C}^N &\to \mathbb{C}^N \\ \boldsymbol{\alpha} &\mapsto \underline{\boldsymbol{\alpha}} \end{aligned}$$

such that k ,

$$\underline{\alpha}_k = \frac{\alpha_k}{|\alpha_k|} \left(|\alpha_k| - \frac{\lambda}{1+\lambda K_{\mathbf{w}_k}} \sum_{\substack{k'=1 \\ k' \in \mathcal{N}(k)}}^{K_k} w_{k'}^{(k)} |\alpha_{k'}| \right)^+ . \quad (5)$$

where $K_{\mathbf{w}_k} = \sum_{\substack{k'=1 \\ k' \in \mathcal{N}(k)}}^{K_k} w_{k'}^{(k)^2}$.

These two heuristic shrinkage operations are computed on singly-indexed coefficients by taking into account their neighborhood. Thus the definition of neighborhood induces a double indexing in the end: one to index the neighborhood, another to index the element which belongs to a neighborhood. The shrinkage is then defined by applying the proximity operator given by the Group-Lasso or the Elitist-Lasso by using this induced double indexing.

3) PE-Lasso: Now, if we consider a set of coefficients which is already doubly indexed by (g, m), we can define two kinds of neighborhoods: a first neighborhood on m and a second neighborhood on g. This will lead to triply-indexed coefficients and we can apply the Elitist-Group-Lasso shrinkage operator, as done above with Group-Lasso and Elitist-Lasso, to obtain the Persistent-Elitist-Lasso [42].

To an index (g, m) in a structured set $\mathcal{I} = \mathcal{I}_g \times \mathcal{I}_m$, we associate a weighted neighborhood $\mathcal{N}(g, m) = \{m' \in \mathcal{I}_m : w_{g,m'}^{(g,m)} \neq 0\}$ with weights $w_{g,m'}^{(g,m)}$ defined on $\mathcal{I}^{|\mathcal{I}|}$, such that $w_{g,m'}^{(g,m)} \geq 0$ for all $(g, m) \in \mathcal{I}$, $m' \in \mathcal{I}_m$, and $w_{g,m}^{(g,m)} > 0$.

For defining the operator, let $[\alpha]_{g,m} \stackrel{\text{def}}{=} \sqrt{\sum_{m' \in \mathcal{N}(g,m)} w_{g,m'}^{(g,m)} \alpha_{g,m'}^2}$. For each g, let the indexing denoted by m'_g be defined such that $\forall m'_g, [\alpha]_{g,m'_g+1} \leq [\alpha]_{g,m'_g}$ and re-order the $[\alpha]$ according to this index. Let the index M_g be such that:

$$\lambda \sum_{m'_g=1}^{M_g} \left([\alpha]_{g,m'_g} - [\alpha]_{g,M_g} \right) < [\alpha]_{g,M_g} \leq \lambda \sum_{m'_g=1}^{M_g+1} \left([\alpha]_{g,m'_g} - [\alpha]_{g,M_g} \right) .$$

Then the Persistent-Elitist-Lasso shrinkage operator [42] is given by

$$\begin{aligned} \mathbb{S}_{\lambda}^{pel}(\boldsymbol{\alpha}) : \mathbb{C}^N &\to \mathbb{C}^N \\ \boldsymbol{\alpha} &\mapsto \underline{\boldsymbol{\alpha}} \end{aligned}$$

where for all g, m ,

$$\underline{\alpha}_{g,m} = \alpha_{g,m} \left(1 - \frac{\lambda}{1+\lambda M_g} \frac{\sum_{m'_g=1}^{M_g} [\alpha]_{g,m'_g}}{\|\boldsymbol{\alpha}_{m' \in \mathcal{N}(g,m)}\|_2} \right)^+ . \quad (6)$$

Here, a coefficient will be selected if its neighborhood is sufficiently energetic compared to the others. Such a structure is illustrated in Fig. 3.

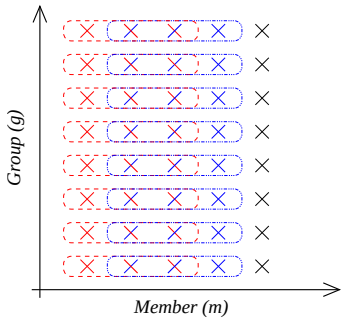

Fig. 3. Persistent Elitist-LASSO. Coefficients are doubly index by (g,m). For each we are considering the left and the right neighbor to define the neighborhood. Then, an E-Lasso selection is done between these groups.

The new operators constructed above are based on the shrinkage/thresholding operators of the proximity operators associated to the convex prior given in Proposition 1. In the present form, they are not directly associated to a convex problem themselves. In the following, we set up an explicit connection between the proximity operators from Proposition 1 and the newly introduced structured shrinkage operators (4) - (6). For this purpose, the next subsection defines a mapping into a bigger space where the proximity operator can be applied.

B. Neighborhoods with latent variables

The neighborhoods, and the groups they implicitly induce, can be formally defined via an expansion operator. This operator maps the original coefficients into a bigger space, where its image consists of copies of the coefficients such that independent groups can be defined over the neighborhood of the coefficients.

Definition 4 (Expansion operator)**.** *Let $\boldsymbol{\alpha} \in \mathbb{C}^N$. Let $\mathcal{E} : \mathbb{C}^N \to \mathbb{C}^{N \times N}$ be an expansion operator such that*

$$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N) \mapsto (\sqrt{w_1^{(1)}}\alpha_1, \ldots, \sqrt{w_N^{(1)}}\alpha_N, \ldots, \sqrt{w_1^{(N)}}\alpha_1, \ldots, \sqrt{w_N^{(N)}}\alpha_N)$$

with $w_i^{(j)} \geq 0$ and $\sum_j w_i^{(j)} = 1$ for all i.

In practice, the non-zeros weight are the same as the weights as in the definition of the social shrinkage operators above. We thus have constructed one group for *each* coefficient, regrouping all its neighborhood by "copying" the coefficients. Due to the condition on the weights, we can state the following proposition which will be crucial later on.

Proposition 2. *$\mathcal{E}$ is isometrical, and $\mathcal{E}^*(\mathcal{E}(\boldsymbol{\alpha})) = \boldsymbol{\alpha}$.*

Proof: Let $\mathcal{E}$ be an expansion operator as defined in Definition 4. Then, for any $\boldsymbol{\alpha} \in \mathbb{C}^N$, we have

$$\begin{aligned}\|\mathcal{E}(\boldsymbol{\alpha})\|^2 &= \sum_i \sum_j |\sqrt{w_i^{(j)}}\alpha_i|^2 = \sum_i |\alpha_i|^2 \sum_j w_i^{(j)} \\ &= \sum_i |\alpha_i|^2 = \|\boldsymbol{\alpha}\|^2 \ .\end{aligned}$$

Hence the proposition. ∎

Let $\mathbf{E}$ denotes the matrix associated with $\mathcal{E}$. This matrix can be viewed as a $N^2 \times N$ matrix where each row contains at most only one non-zero element, corresponding to the weights associated to the elements of a neighborhood:

$$\mathbf{E} = [\mathbf{E}_1, \ldots, \mathbf{E}_N]^T \ , \tag{7}$$

with $\mathbf{E}_i \in \mathbb{R}^{N \times N}$

$$\mathbf{E}_i = [\sqrt{w_1^{(i)}}\mathbf{e}_1^T, \ldots, \sqrt{w_N^{(i)}}\mathbf{e}_N^T]^T \ ,$$

where $\mathbf{e}_j$ is the j^{th} canonical base vector of $\mathbb{R}^N$. A similar expansion matrix has also been used in the context of overlapping groups in [30], [44].

A direct consequence is that one can simply go back to the original space by using the adjoint operator. However, in order to be able to establish a link between the heuristic shrinkage operators previously defined and the common proximity operator, we need to introduce the following left inverse of $\mathbf{E}$:

$$\begin{aligned}\mathbf{D} \ &: \ \mathbb{C}^{N \times N} \to \mathbb{C}^N \\ \mathbf{z} = (z_1^1, \ldots, z_N^1, \ldots, z_1^N, \ldots, z_N^N) &\mapsto \mathbf{x} \\ \text{such that } \forall k, \ x_k &= \frac{1}{\sqrt{w_k^{(k)}}} z_k^k\end{aligned} \tag{8}$$

One can easily check that we have $\mathbf{DE} = I$ and then $\mathbf{ED}$ is a bi-orthogonal (oblique) projection. Moreover $\mathbf{D}$ is such that $\mathbf{DD}^T = \text{diag}\left(\frac{1}{w_k^{(k)}}\right)$. Using these operators, one immediately obtains the following proposition linking the heuristic structured shrinkage and the proximity operators from Section III-A.

Proposition 3. *Let $\mathbb{S}$ be the shrinkage operator of the WG-Lasso* (4)*, WE-Lasso* (5) *or PE-Lasso* (6) *and Ω the regularizer of the G-Lasso, E-Lasso and GE-Lasso, respectively (see Prop. 1). Let $\mathbf{E}$ be the expansion operator* (7) *and $\mathbf{D}$ its left inverse* (8)*. Then*

$$\mathbb{S}_\lambda = \mathbf{D} \circ \text{prox}_{\lambda\Omega} \circ \mathbf{E}$$

Proof: For the sake of brevity, we give the proof for the WG-Lasso, i.e. $\Omega = \|\cdot\|_{21}$. The proofs for WE-Lasso and PE-Lasso are similar.

Thanks to the introduction of the expanded operator $\mathbf{E}$ in Definition 4, we have for $\boldsymbol{\alpha} \in \mathbb{R}^N$ and for a given neighborhood $\mathcal{N}$ on its indices:

$$\sum_{k=1}^N \sqrt{\sum_{\ell \in \mathcal{N}(k)} w_\ell^{(k)} |\alpha_\ell|^2} = \|\mathbf{E}\boldsymbol{\alpha}\|_{21} \ .$$

Then, $\mathbf{z} = \text{prox}_{\lambda\Omega}(\mathbf{E}\boldsymbol{\alpha})$ is given coordinatewise by:

$$\forall \ell \in \mathcal{N}(k), \ z_\ell^k = w_\ell^{(k)} \alpha_k \left(1 - \frac{\lambda}{\sqrt{\sum\limits_{k' \in \mathcal{N}(k)} w_{k'}^{(k)} |\alpha_{k'}|^2}}\right)^+ \ .$$

Therefore, by Definition (8) of $\mathbf{D}$ the claim follows. ∎

Having established the link between social shrinkage and proximity operators, we can construct various algorithms to deal with the problem of "social sparsity".

C. Algorithms for social sparsity

We proceed by introducing heuristic algorithms, based on popular algorithms presented in Section III, in order to embed the previously introduced "social-shrinkage" operators. We then derive a more conventional convex approach thanks to the previously introduced expansion operator, and compare the advantages and shortcomings of the different algorithms.

1) ISTA with social sparsity operator: A natural question is how these operators behave if they are used inside iterative thresholding algorithms. Algorithm 3 rewrites ISTA with a given shrinkage operator $\mathbb{S}$.

Algorithm 3: ISTA with heuristic shrinkage

Initialization: $\boldsymbol{\alpha}^{(0)} \in \mathbb{C}^N$, $k = 1$, $\gamma = \|\boldsymbol{\Phi}\boldsymbol{\Phi}^*\|$
repeat
  $\boldsymbol{\alpha}^{(k)} = \mathbb{S}_{\lambda/\gamma}\big(\boldsymbol{\alpha}^{(k-1)} + \frac{1}{\gamma}\boldsymbol{\Phi}^*(\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha}^{(k-1)})\big)$;
  $k = k + 1$;
until *convergence*;

As the shrinkage operators $\mathbb{S}$ defined in Equations (4), (5) and (6) are not even non-expansive, the convergence study of Algorithm 3 is difficult and remains an open problem. However, experiments show a very good behavior of Algorithm 3 with *any* left inverse of $\mathbf{E}$. Our observations are specified in the following

Conjecture 1. *Let $\mathbb{S}$ be the shrinkage operator of the WG-Lasso* (4)*, WE-Lasso* (5) *or PE-Lasso* (6)*, and set $\gamma = \|\boldsymbol{\Phi}\boldsymbol{\Phi}^*\|$. Let us introduce the operator*

$$T(\boldsymbol{\alpha}) = \mathbb{S}_{\lambda/\gamma}\big(\boldsymbol{\alpha} + \frac{1}{\gamma}\boldsymbol{\Phi}^*(\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha})\big) \ . \tag{9}$$

Then the sequence $\{\boldsymbol{\alpha}^k\}$, generated by Algorithm 3, converges to a fixed point of T.

Moreover, we have also used the same shrinkage operators with FISTA, and have observed the same accelerating effect on the speed of convergence compared to the switch from ISTA to FISTA in the classic convex case.

In the next subsection, we conceptually link the heuristic ISTA with the optimization of a convex functional.

2) Neighborhood as a convex prior: Thanks to the expansion operator $\mathbf{E}$, we can introduce new convex priors which are closely related to our shrinkages, which we denote as cvx-*.

Definition 5 (Social Sparsity Convex Regularizers)**.** *Let* $\boldsymbol{\alpha} \in \mathbb{C}^N$ *and let* $\mathbf{E}$ *be the expansion operator* (7)*.*

cvx-Windowed-Group-Lasso:

$$\begin{aligned}\Omega_{wgl}(\boldsymbol{\alpha}) &= \sum_{k=1}^{N} \sqrt{\sum_{\ell \in \mathcal{N}(k)} w_\ell^{(k)} |\alpha_\ell|^2} \\ &= \|\mathbf{E}\boldsymbol{\alpha}\|_{21} \end{aligned} \tag{10}$$

cvx-Windowed-Elitist-Lasso:

$$\begin{aligned}\Omega_{wel}(\boldsymbol{\alpha}) &= \sum_{k=1}^{N} \left(\sum_{\ell \in \mathcal{N}(k)} \sqrt{w_\ell^{(k)}} |\alpha_\ell| \right)^2 \\ &= \|\mathbf{E}\boldsymbol{\alpha}\|_{12}^2 \end{aligned} \tag{11}$$

cvx-Persistent-Elitist-Lasso:

$$\begin{aligned}\Omega_{pel}(\boldsymbol{\alpha}) &= \sum_{m=1}^{|\mathcal{I}_m|} \left(\sum_{g=1}^{|\mathcal{I}_g|} \sqrt{\sum_{\ell \in \mathcal{N}(g,m)} w_\ell^{(g,m)} |\alpha_\ell|^2} \right)^2 \\ &= \|\mathbf{E}\boldsymbol{\alpha}\|_{212}^2 \end{aligned} \tag{12}$$

A natural convex functional is then

$$F(\boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|^2 + \lambda \Omega(\boldsymbol{\alpha}) \ , \tag{13}$$

where Ω is one of the convex penalty defined above, and one can look for

$$\hat{\alpha} = \operatorname*{argmin}_{\boldsymbol{\alpha} \in \mathbb{C}^N} F(\boldsymbol{\alpha}) \ .$$

Such an approach was also proposed to deal with "overlapping" groups in [30]. Moreover, if all the weights in $\mathbf{E}$ are equal, then we found the mixed norms with overlaps studied in [29]. Notice that the operator $\mathbf{E}$ appears in the penalty, and can then be thought of as an *analysis* prior. Using the results discussed in [45], we can reformulate it as a constrained *synthesis* problem:

$$\hat{\alpha} = \mathbf{E}^T \operatorname*{argmin}_{\mathbf{u}, s.t. \mathbf{u} = \mathbf{E}\mathbf{E}^T\mathbf{u}} \|\mathbf{y} - \boldsymbol{\Phi}\mathbf{E}^T\mathbf{u}\|^2 + \lambda \|\mathbf{u}\|_*$$

where $\| \ \|_*$ is the corresponding (possibly squared) norm used to define the penalty Ω (say $\Omega_{wgl}(\alpha) = \|\mathbf{E}\boldsymbol{\alpha}\|_* = \|\mathbf{E}\boldsymbol{\alpha}\|_{21}$). Interestingly, as $\mathbf{u} = \mathbf{E}\mathbf{E}^T\mathbf{u} \Rightarrow \mathbf{D}\mathbf{u} = \mathbf{E}^T\mathbf{u}$, we have

$$\hat{\alpha} = \mathbf{E}^T \operatorname*{argmin}_{\mathbf{u}, s.t. \mathbf{u} = \mathbf{E}\mathbf{E}^T\mathbf{u}} F_{\mathbf{E}}(\mathbf{u}) = \mathbf{D} \operatorname*{argmin}_{\mathbf{u}, s.t. \mathbf{u} = \mathbf{E}\mathbf{E}^T\mathbf{u}} F_{\mathbf{D}}(\mathbf{u}) \ ,$$

with the two following functionals:

$$F_{\mathbf{E}}(\mathbf{u}) = \|\mathbf{y} - \boldsymbol{\Phi}\mathbf{E}^T\mathbf{u}\|^2 + \lambda \|\mathbf{u}\|_* \tag{14}$$

and

$$F_{\mathbf{D}}(\mathbf{u}) = \|\mathbf{y} - \boldsymbol{\Phi}\mathbf{D}\mathbf{u}\|^2 + \lambda \|\mathbf{u}\|_* \ . \tag{15}$$

The functional (14), which corresponds to a pure synthesis approach, is actually exactly the problem of the latent-Group-Lasso [10], [11].

Seeking an estimate of $\boldsymbol{\alpha}$ as a minimizer of F, it is possible to apply an algorithm as the one proposed in [46], where the proximity operator of the sum of two convex functions is derived from a Douglas Rachford Algorithm. We present such an algorithm with the "ISTA" framework in Algorithm 4, but it can be embedded in FISTA instead. Other approaches such as augmented Lagrangian can also be used (see [30]).

Algorithm 4: Proximal algorithm to minimize F (13).

Initialization: $\boldsymbol{\alpha}^{(0)} \in \mathbb{C}^N$, $k = 1$, $\gamma = \|\boldsymbol{\Phi}\boldsymbol{\Phi}^*\|$
repeat –ISTA loop–
  $\boldsymbol{\alpha}^{k+1/2} = \boldsymbol{\alpha}^{(k)} + \frac{1}{\gamma}\boldsymbol{\Phi}^*(\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha})$;
  $\mathbf{v} = \mathbf{E}\boldsymbol{\alpha}^{k+1/2}$;
  repeat –Douglas-Rachford loop–
    $\mathbf{u} = \mathbf{E}\mathbf{E}^T(\frac{\mathbf{v}+\mathbf{E}\mathbf{y}}{2})$;
    $\mathbf{v} = \mathbf{v} + \operatorname{prox}_{\frac{\lambda}{\gamma}\|\cdot\|_*}(2\mathbf{u} - \mathbf{v}) - \mathbf{u}$
  until *convergence*;
  $\boldsymbol{\alpha}^{k+1} = \mathbf{E}^T\mathbf{u}$;
  $k = k + 1$;
until *convergence*;

Coming back to ISTA with social-shrinkage operators, Algorithm 3, we can rewrite the main iteration as

$$\begin{aligned}&\mathbb{S}_{\frac{\lambda}{\gamma}}(\boldsymbol{\alpha} + \frac{1}{\gamma}\boldsymbol{\Phi}^*(\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha})) = \\ &\qquad \mathbf{D} \operatorname*{argmin}_{\mathbf{u}} \ell_{F_{\mathbf{E}}}(\mathbf{u}, \mathbf{E}\boldsymbol{\alpha}) + \frac{\gamma}{2}\|\mathbf{u} - \mathbf{E}\boldsymbol{\alpha}\|^2 \ .\end{aligned}$$

where $\ell_{F_{\mathbf{E}}}(\mathbf{u}, \mathbf{E}\boldsymbol{\alpha})$ is the linearization of $F_{\mathbf{E}}$ in $\mathbf{E}\boldsymbol{\alpha}$:

$$\ell_{F_{\mathbf{E}}}(\mathbf{u}, \mathbf{E}\boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha}\|^2 + \langle \mathbf{E}\boldsymbol{\Phi}^*(\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha}), \mathbf{u} - \mathbf{E}\boldsymbol{\alpha} \rangle + \|\mathbf{u}\|_* \ .$$

Moreover, using a latent variable $\mathbf{z}$ such that $\boldsymbol{\alpha} = \mathbf{D}\mathbf{z}$, due to the bi-orthogonality of $\mathbf{E}\mathbf{D}$, the main iteration of Algorithm 3 becomes

$$\begin{aligned}\mathbf{z}^{(k)} &= \mathbf{E}\mathbf{D} \ \operatorname{prox}_{\frac{\lambda}{\gamma}\|\cdot\|_*}\left(\tilde{\mathbf{z}}^{(k-1)}\right) \\ \boldsymbol{\alpha}^k &= \mathbf{D}\mathbf{z}^k \\ \text{where } \ \tilde{\mathbf{z}}^{(k-1)} &= \mathbf{z}^{(k-1)} + \frac{\mathbf{E}}{\gamma}\boldsymbol{\Phi}^*(\mathbf{y} - \boldsymbol{\Phi}\mathbf{E}^T\mathbf{z}^{(k-1)})\end{aligned}$$

This can be seen as a the gradient-proximal step followed by an oblique projection.

This remark further leads to the application of the natural left inverse $\mathbf{E}^T$ instead of $\mathbf{D}$ in Proposition 3. We thus obtain a new "WG-Lasso-like" shrinkage operator, the **orth-WG-Lasso**:

$$\begin{aligned}\mathbb{S}_\lambda(\boldsymbol{\alpha}) : \mathbb{C}^N &\to \mathbb{C}^N \\ \boldsymbol{\alpha} &\mapsto \underline{\boldsymbol{\alpha}}\end{aligned}$$

where for all k ,

$$\underline{\alpha}_k = \alpha_k \sum_j w_k^{(j)} \left(1 - \frac{\lambda}{\sqrt{\sum_{j' \in \mathcal{N}(j)} w_{j'}^{(j)} |\alpha_{j'}|^2}} \right)^+ \ . \tag{16}$$

One can see in (16) that in this case, a coefficient will be set to zero only if all the coefficients belonging to its neighborhood are also set to zero.

In this case, Algorithm 3 can be seen as the composition of a proximity operator and an orthogonal projection:

$$\begin{aligned} \mathbf{z}^{(k)} &= \mathbf{E}\mathbf{E}^T \ \mathrm{prox}_{\frac{\lambda}{\gamma}\|\cdot\|_*}\left(\tilde{\mathbf{z}}^{(k-1)}\right) \\ \boldsymbol{\alpha}^k &= \mathbf{E}^T\mathbf{z}^k \\ \text{where } \ \tilde{\mathbf{z}}^{(k-1)} &= \mathbf{z}^{(k-1)} + \frac{\mathbf{E}}{\gamma}\boldsymbol{\Phi}^*(\mathbf{y} - \boldsymbol{\Phi}\mathbf{E}^T\mathbf{z}^{(k-1)}) \end{aligned} \tag{17}$$

As $\mathbf{E}\mathbf{E}^T$ is an orthogonal projection, this algorithm converges to a fixed point by applying [47, Theorem 6.3 and Corollary 6.5].

In summary, the introduction of the social sparsity operators has led to various algorithms related to the minimization of (13). Additionally to Conjecture 1 on the convergence of ISTA in conjunction with the social sparsity operators, we remarked that the social sparsity operators are the composition of an oblique projection and a proximity operator, which makes it possible to instead use an orthogonal projection approach. Thirdly, we noted that (13) can be minimized directly using a Douglas-Rachford algorithm to compute the proximity operator of the corresponding regularizer Ω. Finally, we have established a link to the previously existing methods [10], [29], [30]. Let us stress that using the social sparsity operators (such as WG-Lasso or orth-WG-Lasso) in ISTA as proposed in Algorithm 3 enables to always work directly on the coefficients $\boldsymbol{\alpha}$ without the need of the latent variables. Indeed, an obvious advantage of the WG-Lasso and orth-WG-Lasso over cvx-WG-Lasso and the latent-Group-Lasso is their computational speed. WG-Lasso can be computed in $\mathcal{O}(NK)$ operations where N is the number of coefficients, and K the size of the neighborhood. orth-WG-Lasso has a similar complexity in $\mathcal{O}(2NK)$ while cvx-WG-Lasso must be solved by convex optimization. Moreover, both cvx-WG-Lasso and the latent-Group-Lasso have a fingerprint memory proportional to NK, while their social structured sparse operator WG-Lasso and orth-WG-Lasso have a fingerprint memory proportional to N, independently of the size of the neighborhood.

V. Social sparsity in practice

While the idea of social sparsity was motivated from the stance of time-frequency analysis in the introduction, it can be in fact employed with any type of dictionary. Here, we focus on time-frequency and wavelet-dictionaries and evaluate the developed concepts with respect to audio and image denoising tasks. Although we formally proposed a variety of different shrinkage operators above, their practical exploration will stay confined to the WG-Lasso for the sake of brevity. For applications using other social sparse operators, see [8], [12], [26].

The following numerical evaluation proceeds in three steps. Firstly, the behavior of the WG-Lasso (4) and its counterparts cvx-WGL (10), orth-WGL (16) is characterized by experiments in significance map estimation and denoising employing synthetic signals and a MDCT-dictionary. In order to set our results into context, we additionally include both the basic Lasso and the latent-Group-Lasso in evaluations. Testing for real-life audio signals, it is secondly argued that WGL is a valuable alternative to the state of the art in audio denoising in conjunction with Gabor dictionaries. Finally, it is suggested how to use the social sparse approach with wavelet-dictionaries for image denoising.

Concerning the choice of weights, the necessary condition $\sum_j w_i^{(j)} = 1$ on the weights in order to have the isometrical property and the condition $\sum_i w_i^{(j)} = 1$ given in the definition of the neighborhood are of course totally compatible. The simplest case is when each coefficient is repeated the same number of times, say K, and then the non zero weights are equal to $\frac{1}{K}$. When a "sliding" window is used, the condition is obviously satisfied. Such a window also bears the advantage that the corresponding sums in the persistent shrinkage operators can be computed by fast convolution algorithms. However, it can be interesting to have an asymmetric or a smoother window. A more detailed link between (audio) signal characteristics and optimal neighborhood choice is given in [48]. Here, we rather stick with basic neighborhood shapes in order to clearly present the underlying principles.

Finally, let us note that for all the experiments, we chose to initialize the algorithms with the null vector. We interestingly observed that WG-Lasso and orth-WG-Lasso was scarcely sensitive to the choice of the initialization: we always obtained the same results, independent of the particular initialization.

A MatLab toolbox available at
http://homepage.univie.ac.at/monika.doerfler/StrucAudio.html
provides most of the social shrinkage operators and corresponding algorithms for audio denoising.

A. Time-Frequency Dictionaries

Time-frequency dictionaries as Gabor frames (a.k.a. the short-time Fourier transform) or the MDCT are extensively used in a variety of audio-processing tasks. In order to show the relevance of the social sparsity approach, we perform two kinds of experiments. First, we simulate a signal in order to show the ability of our approach to accurately recover its significance map (i.e. the set of non-zero coefficients). Then, we compare several approaches for a standard denoising problem on real audio signals.

1) Simulations in the orthogonal basis case: Let us assume that $\{\varphi_{(k)}\}_{k=1,\dots,N}$ is a MDCT basis. The following experiment uses such an orthonormal basis with window lengths of 2048 samples. The time-persistent neighborhoods are constructed by setting for any time-frequency index $k = (g, m)$, $\mathcal{N}(k) = \mathcal{N}(g, m) = \{(g-2, m), \dots, (g, m), \dots, (g+2, m)\}$ with g referring to time indices, i.e. each neighborhood comprises two coefficients before and after the centered one.

Here, we consider signals of the form $y = \sum_{k\in\Delta} x_k\varphi_k + b$ where b is an additive Gaussian noise, and Δ is a structured sparse significance map. The latter is generated using fixed frequency Markov chains as introduced in [49], drawing the synthesis coefficients x_k from a standard normal distribution. An example of such a map is displayed in Fig. 4. This produces an overall signal to noise ratio of about 5 dB.

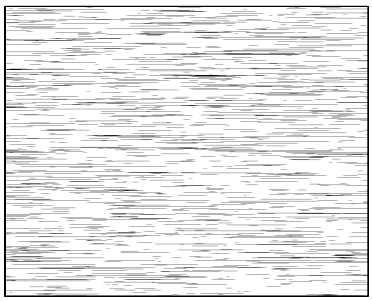

Fig. 4. Time-frequency significance map with 9% non-zero coefficients.

We first compare the ability of various approaches (Lasso, WG-Lasso, orth-WG-Lasso, latent-Group-Lasso and cvx-WG-Lasso) to recover the significance map Δ. For such a task, we use Type 1 and Type 2 errors defined in the following.

Type 1: $\pi_1 = \mathbb{P}\{(t,f) \notin \hat{\Delta} \mid (t,f) \in \Delta\}$;
Type 2: $\pi_2 = \mathbb{P}\{(t,f) \in \hat{\Delta} \mid (t,f) \notin \Delta\}$.

where $\hat{\Delta}$ is the estimated significance map. In other words, Type 1 error refers to the number of false positives, Type 2 error to the number of false-negatives. Fig. 5 depicts the two error types as functions of the estimated significance map size. Clearly, the three structured estimators WG-Lasso, cvx-WG-Lasso and orth-WG-Lasso outperform the non-structured Lasso and the latent-Group-Lasso. While the behavior of cvx-WG-Lasso and orth-WG-Lasso seem to be almost indistinguishable, WG-Lasso appears to perform slightly worse regarding type one errors in a certain sparsity range. However, it exhibits behavior very similar to the two minimization-functional-based estimators for Type 2 errors.

Fig. 6 shows the corresponding denoising results measured in SNR (dB). We choose to display evolution of the SNR versus the number of non-zeros coefficient, and versus the hyperparameter λ. While the WG-Lasso achieves significantly higher SNR than the plain Lasso, its convex counterpart, cvx-WG-Lasso performs slightly better, although only for much higher number of non-zero coefficients. In terms of SNR there does not seem to be a difference between the orth-WG-Lasso and the WG-Lasso, except that the WG-Lasso achieves the same SNR with fewer coefficients. In conclusion, the experiments using signals generated from structured significance maps demonstrate that the WG-Lasso behaves not identically, but quite similar to its convex and orthonormal counterparts. It thus appears as an efficient alternative to these operators, tractable for many real-life applications.

In light of Fig. 7, it is visible that the cardinality of the social sparsity maps presents a fast transition with respect to hyperparameter λ, compared to the Lasso approach. And even for the latent-Group-Lasso, the size of the estimated significance map starts to decrease for smaller λ. Moreover, as shown on the top of Fig. 6, the SNR of the three social-sparsity approaches reaches its maximum quickly and then decreases slowly. A practical implication of these remarks is that the hyperparameter tuning for social sparsity operators might be more straight-forward than in the Lasso case, since

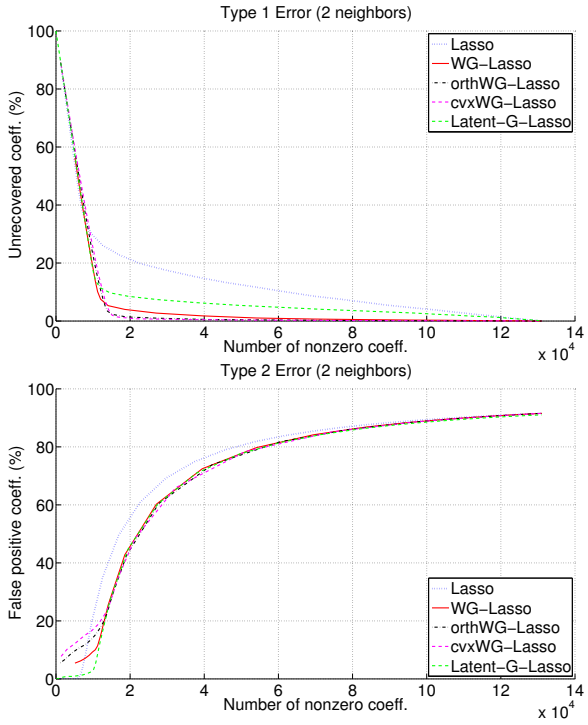

Fig. 5. Top: type 1 error. Bottom: type 2 error.

a fairly good SNR can be reached by choosing λ within the transition phase, which shows a sharp and thus more determined behaviour.

The performances of the latent-Group-Lasso may appear slightly disappointing here. However, we must insist that in the proposed experiments, we used a sliding window in time to construct the neighborhood. In this situation, the *discarding versus selecting* problem, which was the main motivation of the latent-group-lasso, is less important. Indeed, in this particular situation, configurations obtained by discarding groups or selecting them can be very close.

2) Real signals and the overcomplete case: We proceed to demonstrate the benefits of the social sparse approach for audio denoising, using "real-life" signals and an overcomplete Gabor-dictionary. The latter dictionary was also employed by the state of the art in audio denoising, namely the *Block-Thresholding* algorithm [50]. To compare these approaches, we use WG-Lasso with a neighborhood extending over time with 4 coefficients before and after the center coefficient and employ a tight Gabor-frame with Hann-window of length 1024 samples and overlap 4. The chosen test signal is a 6-sec excerpt of a Jazz-quintet producing a complex mixture of drums, double-bass, piano, saxophone and trumpet. In [50], the variance of the noise is supposed to be known, but we choose here to use its value as a parameter of the method. Then, all algorithms depend on only one parameter, which can be tuned according to the variance of the noise.

Fig. 8 shows the corresponding denoising results measured in SNR as functions of the hyperparameter λ for ground-noise levels of 0 and 20 dB SNR. While the Lasso performs constantly worse, Block-Thresholding performs better than WG-Lasso for the 0 dB noise level, and vice versa for the 20

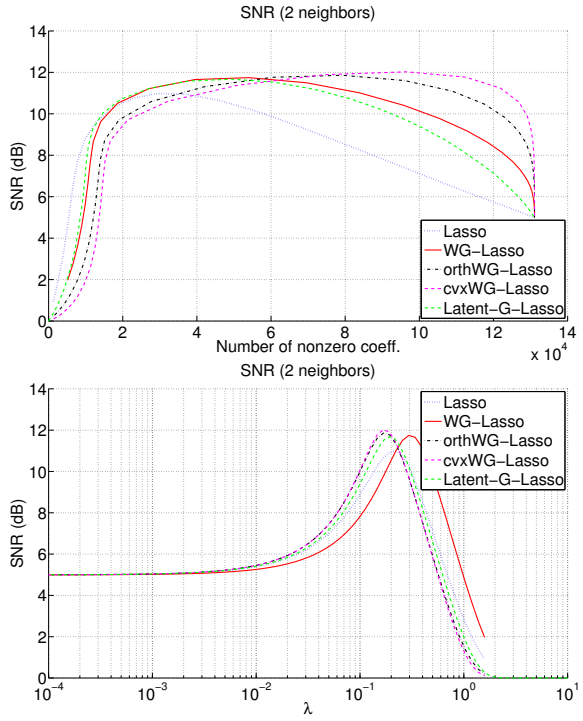

Fig. 6. Top: snr vs non zeros coeff. Bottom: snr vs hyperparameter λ

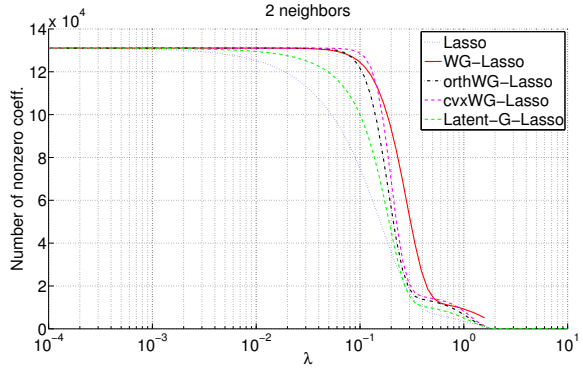

Fig. 7. Evolution of the cardinality of the estimated map

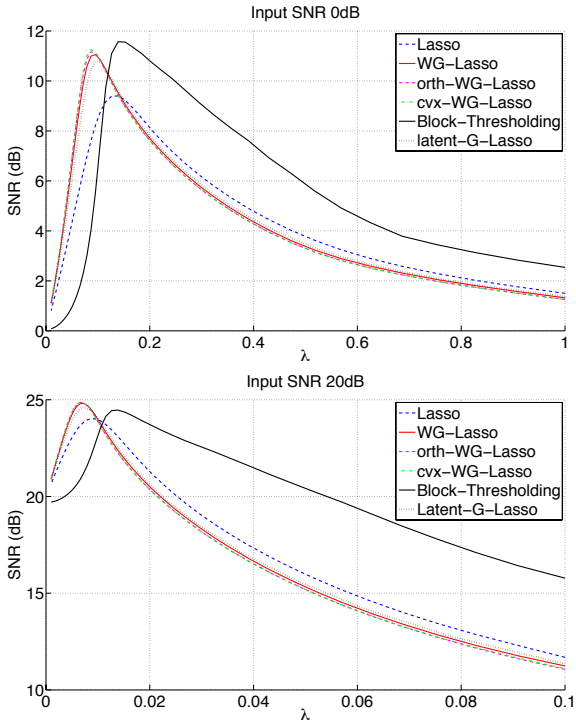

Fig. 8. SNR vs hyperparameter λ for the six operators Lasso, WG-Lasso, orth-WG-Lasso, cvx-WG-Lasso, latent-G-Lasso and Block-Thresholding using a 6 sec complex audio signal containing drums, double-bass, piano, saxophone and trumpet.

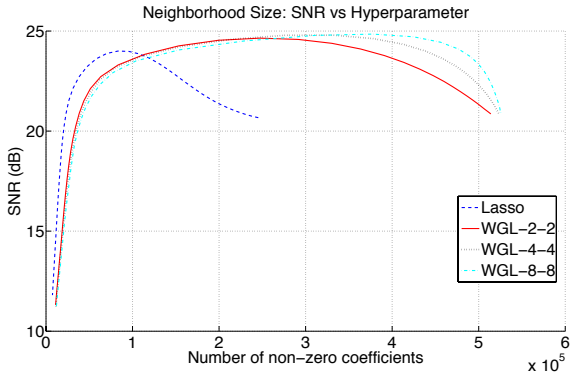

Fig. 9. Comparison of different neighborhood sizes for the WGL operator. Each neighborhood extends in time with 0 (Lasso case), 2, 4 and 8 coefficients before and after the center index.

dB case. Let us note that WG-Lasso, orth-WG-Lasso and cvx-WG-Lasso perform almost identically. The latent-Group-Lasso performs a bit worse than the other structured methods. We also have noticed that the dependence between the size of the maps and the hyperparameter λ is similar to our observations in the simulated case. In particular, WG-Lasso reaches its maximum for sparser significance maps than the orth-WG-Lasso and cvx-Lasso.

For the same signal and noise level of 20 dB, Fig. 9 compares different sizes of neighborhoods (each extending in time). It seems clear that the algorithm is relatively robust w.r.t. the choice of the neighborhood: there is a significant increase in performance from the Lasso to WGL with 2 coefficients in time, before and after the center, but further enlarging the neighborhood does not change results dramatically.

It has to be noted that in terms of SNR, block-thresholding seems to be favorable for some other signals and noise levels we tested. On the contrary, the social sparse approach bears many important advantages over block-thresholding: It is computationally much more efficient (in our setting around a factor 2, obviously depending on the number of iterations) and seems to provide perceptually preferable results, cf. [48]. It should also be noticed that no post-processing was executed here on the results given by the various *-Lasso approaches, and in particular, we did not perform any Wiener estimate contrary to the Block-Thresholding algorithm. Such a post-processing can increase the SNR [50], but we choose to show the raw results of the *-Lasso.

From a computational point of view, cvx-WG-Lasso is very time-consuming and intractable in practice. Fig. 10 depicts the evolution of the value of the functional (13) over the number of iterations for cvx-WG-Lasso, and the FISTA and ISTA

versions of the orth-WG-Lasso and WG-Lasso for the 20 dB case and $\lambda \simeq 10^{-3}$. cvx-WG-Lasso was implemented with ISTA with only one iteration of the Douglas-Rachford inner loop: this strategy appeared to be the most efficient to obtain our results; in particular, FISTA was divergent in practice because of its sensibility to the error done to approximate the proximity operator with the Douglas-Rachford inner loop.

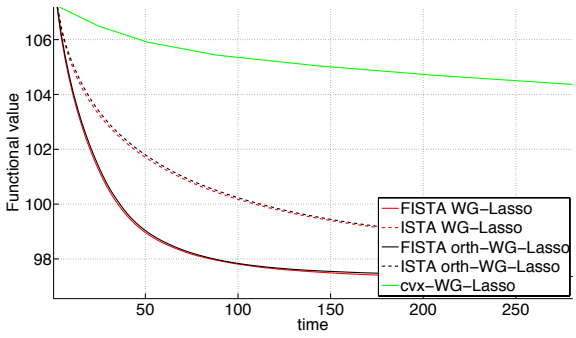

Fig. 10. Functional (13) value with respect to the time for the five algorithms used.

We close the practical investigation of audio denoising with a comparison of the time-frequency maps given by the Lasso, WG-Lasso and orth-WG-Lasso in the 20 dB case for $\lambda \simeq 10^{-3}$. One can observe in Fig. 11 very structured time-frequency maps for WG-Lasso and orth-WG-Lasso compared to the Lasso, and in particular their ability to keep high frequency coefficients. Moreover, it can be seen, that sparsity is more expressed in the map obtained with WG-Lasso; this observation confirms the results shown Fig.10, namely, that WG-Lasso promotes sparser representations than orth-WG-Lasso.

B. Wavelet Dictionaries

While the proposed approach might be particularly intuitive for exploiting persistence in time-frequency representations, it turns out to be similarly promising in conjunction with wavelet dictionaries for applications in image processing, for instance. Here, structures such as sharp edges are sparsely represented, but at the same time exhibit persistence properties along the wavelet tree: if a given coefficient is active, it is highly likely that its respective "father" is so, as well. We hence explore the usage of an asymmetric neighborhood system which emphasizes this directed relation between "father and son", as depicted in Fig. 12.

Notice that the idea of taking into account the persistence of the wavelet coefficients along the tree is not new. Several works propose to modelize this case of "structured sparsity" such as [39], [40], [51].

Initial experiments on image denoising were conducted using the well-known Lena-image to which a Gaussian white noise was added, yielding a peak signal to noise ratio (PSNR) of 20 dB. Fig. 13 compares the performance of the WGL-operator using the described neighborhood system with the plain Lasso estimate, depicting PSNR of the reconstruction as function of the number of non-zero coefficients. Clearly,

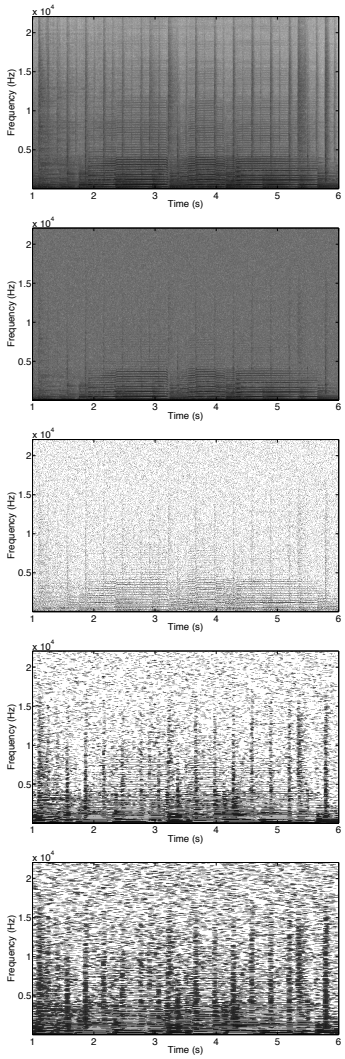

Fig. 11. Time-Frequency maps. From top to bottom: original signal, noisy signal, Lasso, WG-Lasso, orth-WG-Lasso

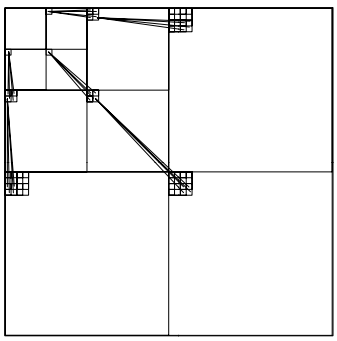

Fig. 12. Asymmetric neighborhoods of wavelet coefficients: each coefficients at scale $j-1$ is grouped with its father at scale j.

WG-Lasso outperforms the Lasso leading to a gain in SNR of about 1 dB.

VI. Conclusion

Social sparsity operators allow to shrink dictionary coefficients with respect to their weighted neighborhoods. In summary, three different but related approaches were presented:

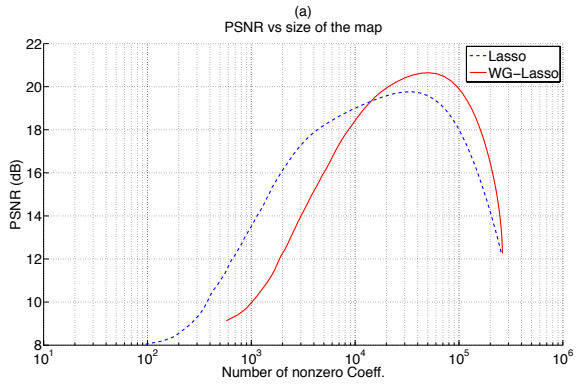

Fig. 13. Top: snr vs non zeros coeff. Bottom: snr vs hyperparameter λ

the oblique projected proximal algorithm, an orthogonal projected proximal algorithm and the minimization of a convex functional.

While the convergence of the last two approaches can be proven, the convergence of the oblique projected proximal algorithm remains an open problem. Moreover, the algorithm can be accelerated with FISTA as for classical convex optimization. These approaches are complemented by the latent-Group-Lasso which corresponds to the unconstrained synthesis version of the proposed convex functional. In particular for WG-Lasso, various experiments on both simulated and real signals demonstrate a very good behavior in denoising tasks: its performance is comparable to the state of the art Block-Thresholding algorithm but it is considerably faster.

Further studies will be devoted to two aspects. On the one hand, a theoretical study of the oblique-projected proximal algorithm must be conducted. On the other hand, we plan to build a practical audio restoration algorithm, using social sparsity operators as the WG-Lasso with hybrid decompositions [4] and a few instinctive, or ideally none, hyperparameters.

ACKNOWLEDGMENT

The authors would like to warmly thank Bruno Torrésani for fruitful discussions and his previous advice. The authors further express their gratitude to the anonymous reviewers for their constructive and valuable remarks.

REFERENCES

[1] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[2] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society Serie B*, vol. 58, no. 1, pp. 267–288, 1996.

[3] R. Rubinstein, A. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of IEEE - Special Issue on Applications of Sparse Representation & Compressive Sensing*, vol. 98, no. 6, pp. 1045–1057, 2010.

[4] L. Daudet and B. Torrésani, "Hybrid representations for audiophonic signal encoding," *Signal Procesing*, vol. 82, no. 11, pp. 1595–1617, 2002.

[5] M. Elad, J.-L. Starck, D. L. Donoho, and P. Querre, "Simultaneous cartoon and texture image inpainting using morphological component analysis (mca)," *Journal on Applied and Computational Harmonic Analysis*, vol. 19, pp. 340–358, November 2005.

[6] R. Gribonval and M. Nielsen, "Sparse representations in unions of bases," *IEEE Transactions on Information Theory*, vol. 49, no. 12, pp. 3320–3325, 2003.

[7] M. Kowalski and B. Torrésani, "Sparsity and persistence: mixed norms provide simple signals models with dependent coefficients," *Signal, Image and Video Processing*, vol. 3, no. 3, pp. 251–264, 2008.

[8] K. Siedenburg and M. Dörfler, "Structured sparsity for audio signals," in *Proceeding of 14th conference on digital audio effects (DAFx)*, Paris, France, September 2011.

[9] M. Kowalski, "Sparse regression using mixed norms," *Applied and Computational Harmonic Analysis*, vol. 37, no. 3, pp. 303–324, 2009.

[10] L. Jacob, G. Obozinski, and J.-P. Vert, "Group lasso with overlap and graph lasso," in *ICML*, 2009.

[11] G. Obozinski, L. Jacob, and J.-P. Vert, "Group lasso with overlaps: the latent group lasso approach." Tech. Rep., 2012.

[12] M. Kowalski, E. Vincent, and R. Gribonval, "Beyond the narrowband approximation: Wideband convex methods for under-determined reverberant audio source separation," *IEEE Transactions on Audio Speech and Language Processing, Special Issue on: "Processing Reverberant"*, vol. 17, no. 7, pp. 1818–1829, 2010.

[13] M. Osborne, B. Presnell, and B. Turlach, "On the lasso and its dual," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 319–337, Jun 2000.

[14] J. A. Tropp, "Just relax: Convex programming methods for identifying sparse signals," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 1030–1051, Mar. 2006.

[15] J.-J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bull. Soc. Math. France*, vol. 93, pp. 273–299, 1965.

[16] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413 – 1457, August 2004.

[17] M. Figueiredo and R. Nowak, "An em algorithm for wavelet-based image restoration," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 906–916, 2003.

[18] P. Combettes and V. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1168–1200, Nov. 2005.

[19] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[20] P. Tseng, "Approximation accuracy, gradient methods, and error bound for structured convex optimization," *Mathematical Programming*, vol. 125, pp. 263–295, 2010, 10.1007/s10107-010-0394-2.

[21] A. Benedek and R. Panzone, "The space l^p with mixed norm," *Duke Mathematical Journal*, vol. 28, pp. 301–324, 1961.

[22] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society Serie B*, vol. 68, no. 1, pp. 49–67, 2006.

[23] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2477–2488, July 2005.

[24] M. Fornasier and H. Rauhut, "Recovery algorithm for vector-valued data with joint sparsity constraints," *SIAM Journal on Numerical Analysis*, vol. 46, no. 2, pp. 577–613, 2008.

[25] Y. Zhou, R. Jin, and S. C. Hoi, "Exclusive lasso for multi-task feature selection," in *Proceeding of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Sardinia, Italy, May 2010.

[26] A. Gramfort and M. Kowalski, "Improving m/eeg source localization with an inter-condition sparse prior," in *ISBI*, Boston, USA, 2009.

[27] M. Szafranski, Y. Grandvalet, and P. Morizet-Mahoudeaux, "Hierarchical penalization," in *Advances in Neural Information Processing Systems 20 (NIPS)*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008.

[28] J.-J. Fuchs, "Extension of the global matched filter to structured groups of atoms: Application to harmonic signals," in *Proc. International Conference on Audio Speech and Signal Processing (ICASSP)*, Praga, Czech-Republic, May 2011.

[29] I. Bayram, "Mixed norms with overlapping groups as signal priors," in *Proc. International Conference on Audio Speech and Signal Processing (ICASSP)*, Praga, Czech-Republic, May 2011.

[30] G. Peyré and J. Fadili, "Group sparsity with overlapping partition functions," in *Eusipco'11*, Spain, 2011.

[31] A. Argyriou, R. Foygel, and N. Srebro, "Sparse prediction with the k-support norm," in *Neural Information Processing Systems (NIPS)*, Lake Tahoe, Nevada, USA, 2012.

[32] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society Series B*, Jan 2005.

[33] S. Mosci, S. Villa, A. Verri, and L. Rosasco, "A primal-dual algorithm for group sparse regularization with overlapping groups," in *Advances in Neural Information Processing Systems 20 (NIPS)*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Kaufmann, 2010, pp. 2604–2612.

[34] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for hierarchical sparse coding," *Journal of Machine Learning Research*, vol. 12, pp. 2297–2334, 2011.

[35] P. Sprechmann, I. Ramírez, G. Sapiro, and Y. C. Eldar, "C-hilasso: A collaborative hierarchical sparse modeling framework." *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4183–4198, 2011.

[36] A. Gramfort, D. Strohmeier, J. Haueisen, M. Hmlinen, and M. Kowalski, "Functional brain imaging with m/eeg using structured sparsity in time-frequency dictionaries," in *IPMI*, Germany, 2011.

[37] R. Jenatton, J.-Y. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *Journal of Machine Learning Research*, vol. 12, pp. 2777–2824, 2011.

[38] C. A. Micchelli, J. M. Morales, and M. Pontil, "A family of penalty functions for structured sparsity," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, 2010.

[39] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, pp. 1982–2001, 2010.

[40] J. Huang, T. Zhang, and D. Metaxas, "Learning with structured sparsity," in *Proceedings of the International Conference on Machine Learning (ICML)*, Montreal, Canada, 2009.

[41] Z. Zhang and B. D. Rao, "Exploiting correlation in sparse signal recovery problems: Multiple measurement vectors, block sparsity, and time-varying sparsity," in *ICML 2011 Workshop on Structured Sparsity: Learning and Inference*, Bellevue, Washington, USA, 2011.

[42] M. Kowalski and B. Torrésani, "Structured sparsity: from mixed norms to structured shrinkage," in *Proceeding of Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, Saint-Malo, France, April 2009.

[43] T. Cai and B. W. Silverman, "Incorporating information on neighbouring coefficients into wavelet estimation," *shankhya: the indian journal of statistic. Special issue on wavelets. Series B.*, vol. 63, pp. 127–148, 2001.

[44] X. Chen, Q. Lin, S. Kim, and J. Carbonell, "Smoothing proximal gradient method for general structured sparse learning," in *Uncertainty in Artificial Intelligence Conference (UAI)*, Barcelona, Spain, Jul. 2011.

[45] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, pp. 947–968, June 2007.

[46] C. Chaux, J.-C. Pesquet, and N. Pustelnik, "Nested iterative algorithms for convex constrained image recovery problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 730–762, 2009.

[47] P. Combettes, "Solving monotone inclusions via compositions of nonexpansive averaged operators," *Optimization*, vol. 53, no. 5-6, pp. 475–504, Dec. 2004.

[48] K. Siedenburg and M. Dörfler, "Audio denoising by generalized time-frequency thresholding," in *Proceedings of the AES 45th Conference on Applications of Time-Frequency Processing in Audio*, Helsinki, Finland, March 2012.

[49] S. Molla and B. Torresani, "A hybrid scheme for encoding audio signal using hidden markov models of waveforms," *Applied and Computational Harmonic Analysis*, vol. 18, no. 2, pp. 137–166, 2005.

[50] G. Yu, S. Mallat, and E. Bacry, "Audio denoising by time-frequency block thresholding," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1830–1839, 2008.

[51] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, 1993.